



Universidade do Porto

Faculdade de Engenharia

FEUP

Eficiência Energética em Redes de Sensores sem Fios

Agentes e Inteligência Artificial Distribuída - 2015/16

Grupo T11_3:

Filipe Ribeiro	201104129	ei11141@fe.up.pt
Henrique Ferrolho	201202772	ei12079@fe.up.pt
João Pereira	201104203	ei12023@fe.up.pt

13 de Dezembro de 2015

Índice

[Índice](#)

[Objectivo](#)

[Descrição do cenário](#)

[Objectivo do trabalho](#)

[Especificação](#)

[Identificação e caracterização dos agentes](#)

[Arquitectura](#)

[Comportamento](#)

[COSA: Comportamento simplificado de um agente](#)

[Protocolos de interacção](#)

[Desenvolvimento](#)

[Plataforma e ferramentas utilizadas](#)

[JADE](#)

[REPAST 3](#)

[SAJaS](#)

[MASSim2Dev](#)

[Ambiente de desenvolvimento](#)

[Estrutura da aplicação](#)

[Diagrama de classes](#)

[Detalhes relevantes da implementação](#)

[GUI](#)

[Experiências](#)

[Experiência 1](#)

[Resultado](#)

[Experiência 2](#)

[Resultado](#)

[Experiência 3](#)

[Resultado](#)

[Conclusões](#)

[Análise dos resultados das experiências](#)

[Desenvolvimento do trabalho](#)

[Melhoramentos](#)

[Recursos](#)

[Bibliografia](#)

[Software](#)

[Elementos do grupo](#)

Objectivo

Descrição do cenário

No âmbito da unidade curricular de Agentes e Inteligência Artificial Distribuída, o grupo propôs-se a desenvolver um programa que possibilita a redução do gasto de energia numa Wireless Sensor Network (WSN).

O problema surge aquando da instalação de uma WSN em áreas inacessíveis ou remotas, em que é necessário o uso de algoritmos de percepção e comunicação que minimizem o gasto de energia. Este gasto é um contratempo devido ao elevado custo que a substituição e recarga das baterias dos sensores têm.

Por outro lado, a informação sensorial transmitida tem de ter qualidade suficiente de forma a permitir a tomada de decisões atempada. Há, portanto, que balancear a qualidade da informação com o consumo energético.

Objectivo do trabalho

O principal objetivo deste trabalho é a implementação e análise dos resultados experimentais de um algoritmo de formação de coligações: *Coalition Oriented Sensing Algorithm* - **COSA**.

As coligações são estabelecidas através de negociações *peer-to-peer* entre os agentes. A estrutura de coligação resultante irá depender do estado dos agentes e do ambiente a cada instante.

Idealmente, o algoritmo aumenta a duração do tempo de vida global dos sensores, causando o mínimo impacto na qualidade da informação recolhida.

Especificação

Identificação e caracterização dos agentes

A aplicação desenvolvida apenas precisa de um tipo de agente, que é instanciado múltiplas vezes. Esse agente é o **SensingAgent**.

Arquitectura

Cada *SensingAgent* é constituído pelas seguintes partes:

- **x, y** - coordenadas da sua localização no rio
- **state** - *on*, *sleeping*, ou *off*
- **sleepCountdown** - contador decrescente para este agente despertar da hibernação
- **energyLevel** - nível actual de energia/bateria
- **color** - cor utilizada para identificar a coligação liderada por este agente
- **stdDeviation** - desvio padrão deste agente
- **maxAdherence** - *aderência* máxima registada
- **maxLeadership** - *liderança* máxima registada
- **pollutionSamples** - lista com as amostras recolhidas por este agente desde o início da simulação
- **leader** - booleano utilizado para identificar este agente como líder, ou dependente
- **neighboursLastSampleMap** - estrutura usada para mapear a última amostragem recolhida por cada vizinho deste agente
- **neighboursAdherenceMap** - estrutura usada para mapear a aderência de cada vizinho deste agente
- **dependantNeighbours** - lista de vizinhos dependentes da coligação liderada por este agente
- **nodeLeaderOfMe** - identificador do agente que lidera a coligação à qual eu pertenço

Comportamento

O algoritmo COSA implica que os nós da rede - os **agentes** do sistema multi-agentes deste projeto - tenham um comportamento padrão de amostragem diferente do normal. O algoritmo implica que os agentes sejam **autónomos**, **proativos** e **reativos**. De acordo com o material utilizado nas aulas teóricas de AIAD, este tipo de arquitetura enquadra-se na de **agentes reativos simples**.

O comportamento dos agentes descrito no parágrafo acima é possível através de um protocolo de negociação simples, e de duas funções de modelação de relações entre os nós: **aderência** e **liderança**.

Os resultados destas funções de modelação de relações determinam o diálogo assíncrono que se dá enquanto os agentes estão a negociar entre eles. Por sua vez, os resultados dessas negociações modificam as funções de modelação de relações.

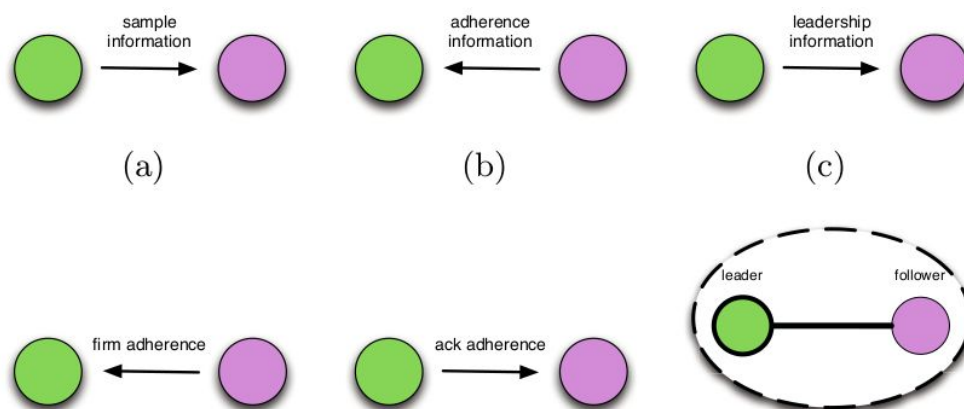
COSA: Comportamento simplificado de um agente

```
while (bateria > 0) {  
    recolheAmostra();  
    atualizaModelo();  
    atualizaRelaçãoComVizinhos();  
    atualizaRedeDeNós();  
}
```

Protocolos de interação

Quando um nó faz uma amostragem do ambiente à sua volta, envia o valor observado para os seus vizinhos. Quando um nó recebe uma amostra de um dos seus vizinhos, usa essa informação para avaliar a pertinência de formar um grupo. Se essa avaliação for positiva, o agente comunica o resultado ao nó que lhe enviou a amostra, e assume o papel de líder. Se ambos concordarem, o agente assume definitivamente o papel de líder, enquanto que o outro assume o papel de dependente. O agente dependente pode hibernar e interromper tanto o seu processo de amostragem, bem como a comunicação com o *sink node*.

Decorrido algum tempo, as negociações simples entre agentes vizinhos acabam por originar uma coligação (através de uma abordagem *bottom-up*) e selecionar o papel mais adequado nessa coligação.



A organização dos nós em grupos bem estruturados traduz-se numa poupança de energia devido à poupança de transmissões de longa distância: aquelas que os nós dependentes não têm que fazer.

Algorithm 2. Message Processing

Data: me : focus node; a_j : generic neighbour; a_l : potential leader; a_r : potential dependant on me; a_p : dependant node on me; a_L : leader node of me;
 $D(me)$: set of dependant nodes on me

```
1 case  $rcvd(inform(a_j, me, meas, t))$ 
2    $updateNeighbourInfo();$ 
3    $adherence2NeighbourEvaluation();$ 
4    $updateOwnMaxAdherence();$ 
5   if  $changesOnOwnMaxAdherence$ 
6     then
7        $inform(me, a_l, maxAdh, t);$ 
8     end
9 case  $rcvd(inform(a_j, me, maxAdh, t))$ 
10   $inform(me, a_r, lead);$ 
11   $updateNeighbourInfo();$ 
12   $adherence2NeighbourEvaluation();$ 
13   $updateOwnMaxAdherence();$ 
14  if  $changesOnOwnMaxAdherence$ 
15    then
16       $inform(me, a_l, maxAdh, t);$ 
17    end
18 case  $rcvd(inform(a_l, me, lead))$ 
19   if  $checkAgainstOwnLead$  then
20      $firmAdherence(me, a_l);$ 
21   end
22 end

23 case  $rcvd(firmAdherence(a_r, me))$ 
24   if  $checkAgainstOwnLead$  then
25      $ackAdherence(me, a_r);$ 
26      $updateOwnLeadValue();$ 
27      $updateDependentGroup();$ 
28   end
29 end
30 case  $rcvd(ackAdherence(a_l, me))$ 
31   if  $!leader \wedge a_l! = a_L$  then
32      $withdraw(me, a_L);$ 
33   end
34   if  $leader \wedge D(me)! = \emptyset$  then
35     while  $D(me)! = \emptyset$  do
36        $break(me, a_p);$ 
37     end
38   end
39    $updateRoleState(dependant);$ 
40    $sleep(t);$ 
41 end
42 case  $rcvd(break(a_L, me))$ 
43    $updateRoleState(leader);$ 
44 end
45 case  $rcvd(withdraw(a_p, me))$ 
46    $D(me) \leftarrow D(me) \setminus a_p;$ 
47    $updateRoleState(leader);$ 
48 end
```

Desenvolvimento

Plataforma e ferramentas utilizadas

JADE

Esta estrutura *open source* tem como objetivo simplificar o desenvolvimento de sistemas multi-agentes, de acordo com as especificações da *FIPA (Foundation for Intelligent Physical Agents)*. O *JADE* tem um sistema multi-agente distribuído, isto é, os agentes apenas podem agir dentro do seu domínio. Cada agente tem *behaviours*, que são tarefas a executar. Estas podem ser executadas concorrentemente e podem ser agendadas de uma forma cooperativa. O *JADE* permite a mobilidade e cloning de agentes.

REPAST 3

O *REPAST 3* é utilizado para fazer simulações com agentes. Ao contrário de *JADE*, a sua finalidade não é desenvolver sistemas multi-agentes. Também não é compatível com as especificações da *FIPA*. Com o *REPAST 3* é possível fazer a recolha de dados e a sua consequente exposição. É ainda possível ver a interação entre os agentes enquanto as simulações estão a ser executadas.

SAJaS

O *SAJaS* é uma API desenvolvida para colmatar a falha entre a simulação e o desenvolvimento de sistemas multi-agentes.

MASSim2Dev

A ferramenta *MASSim2Dev* é um complemento ao *SAJaS*. Possibilita a conversão automática de simulações baseadas em *SAJaS* para *JADE*, e vice-versa.

Ambiente de desenvolvimento

O trabalho foi desenvolvido no sistema operativo **Ubuntu 15.10**, usando o IDE **Eclipse Mars.1**.

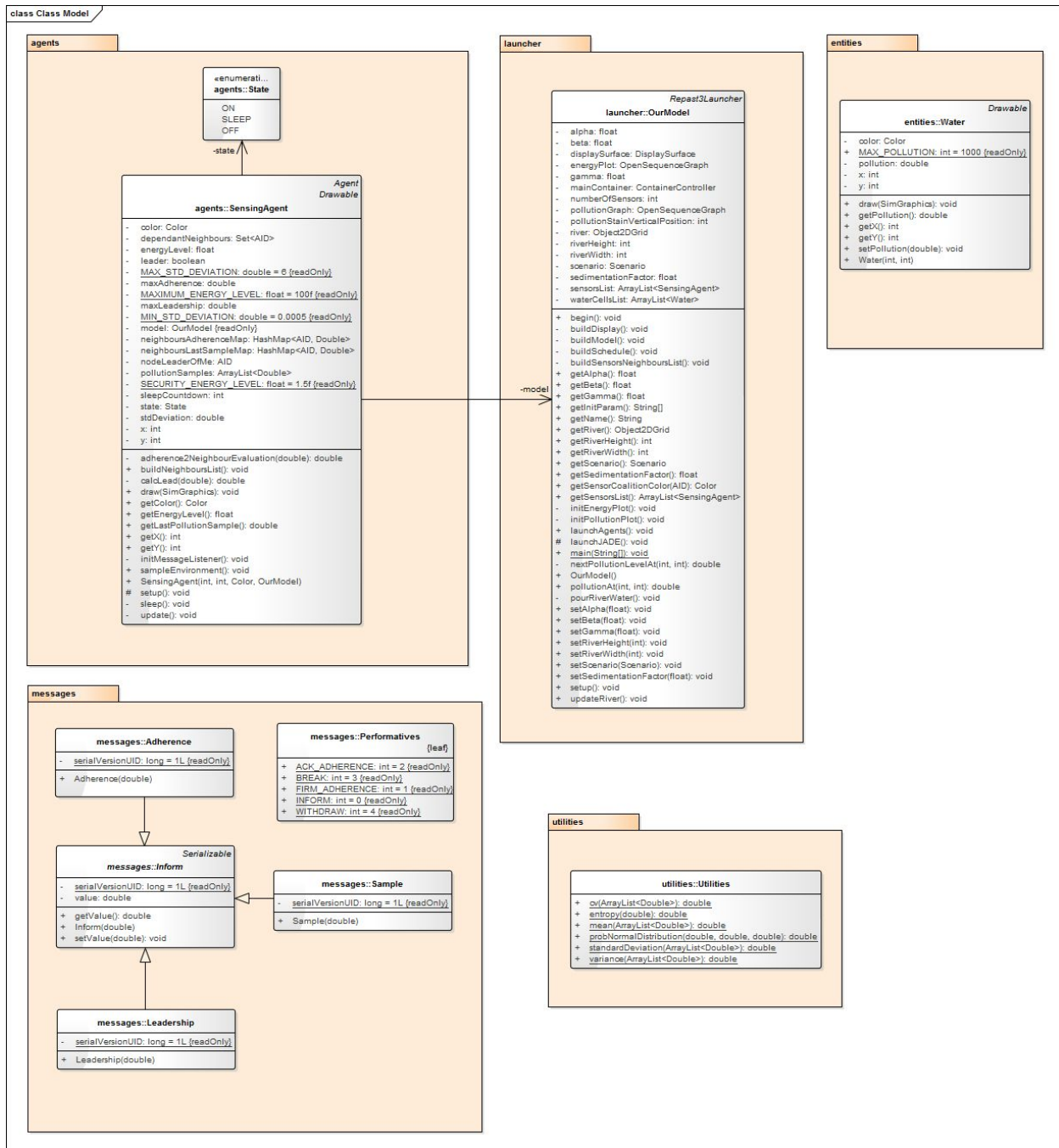
Estrutura da aplicação

A aplicação está estruturada em **cinco pacotes/módulos**:

- **agents**
 - *class SensingAgent* - classe dos agentes da simulação

- *enum State* - enumerador com os estados que um sensor pode tomar
- **entities**
 - *class Water* - classe usada para instanciar cada célula de água do rio
- **launcher**
 - *class OurModel* - classe que representa o modelo da simulação
- **messages**
 - *abstract class Inform*
 - *class Adherence* - classe usada para enviar mensagens com valores de aderência
 - *class Leadership* - classe usada para enviar mensagens com valores de liderança
 - *class Sample* - classe usada para enviar mensagens com valores de amostras de poluição recolhidas no rio
 - *final class Performatives* - classe com constantes usadas para identificar as *performatives* das mensagens enviadas nas comunicações *peer-to-peer* entre os agentes
- **utilities**
 - *class Utilities* - classe com métodos estáticos genericamente classificados como úteis para o processamento de eventos durante as simulações

Diagrama de classes



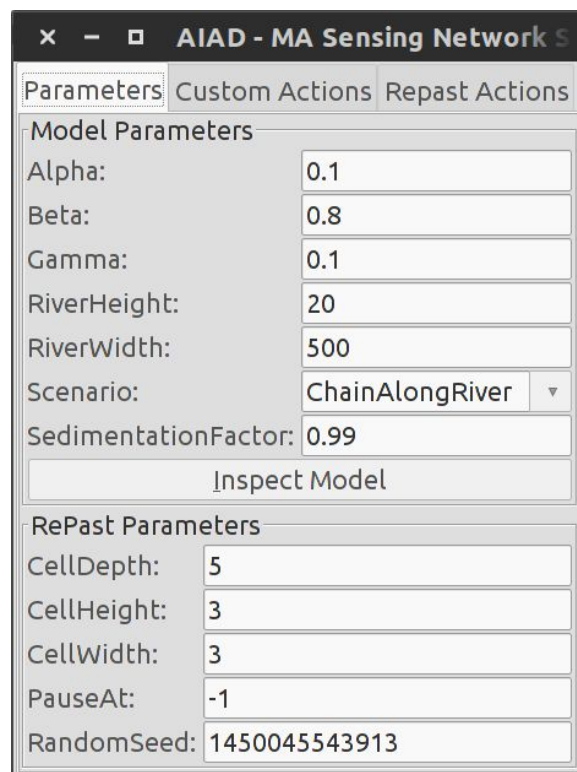
Detalhes relevantes da implementação

GUI

A interface gráfica da aplicação é composta por cinco janelas diferentes. Quando a aplicação é iniciada, apenas duas dessas janelas são visíveis.

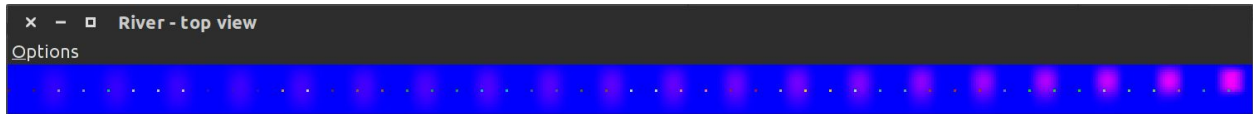


Esta é a janela dos controlos da simulação. É possível fazer *Play*, *Stop*, *Pause*, *Step*, e parar a simulação. É ainda possível ver a contagem dos ticks desde o início da simulação.

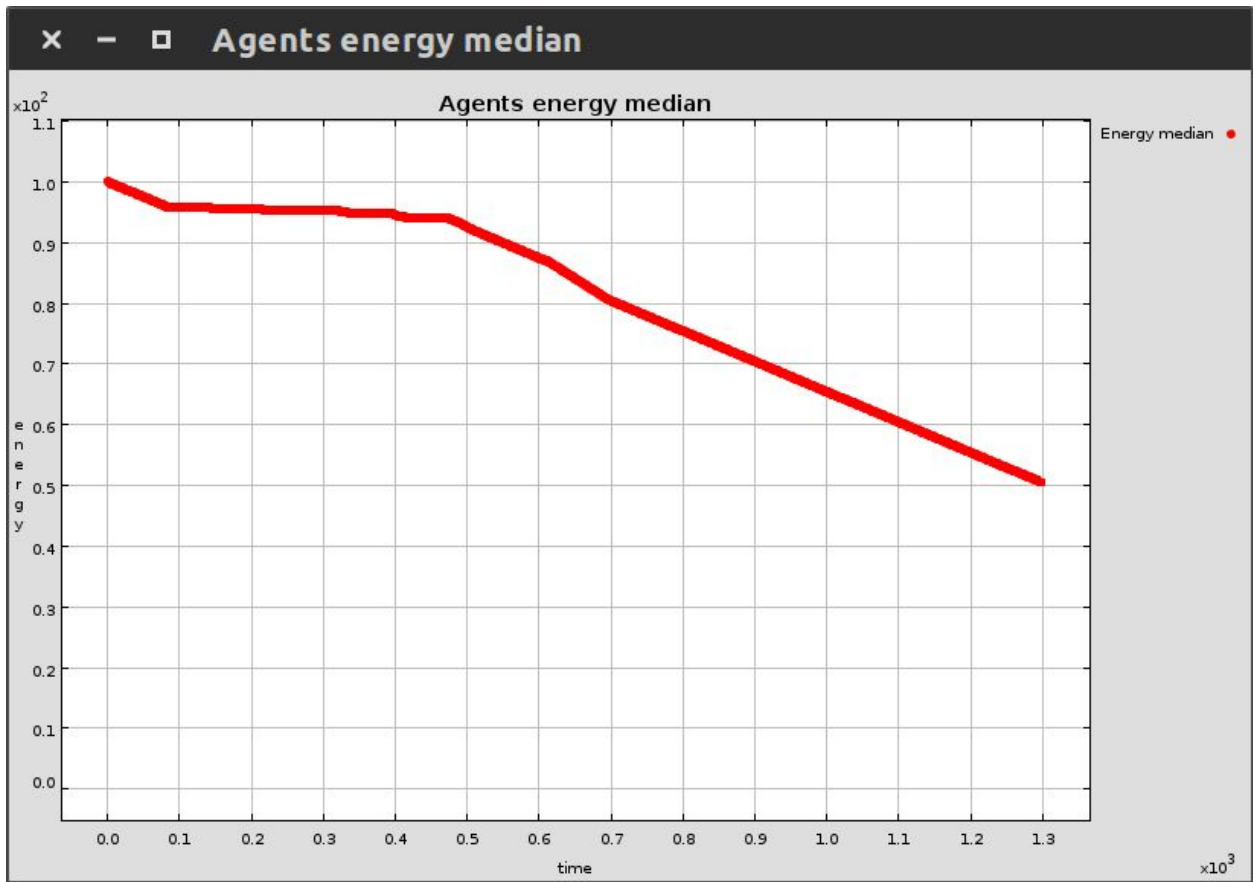


Esta janela contém os parâmetros configuráveis que são usados durante a simulação. Antes de iniciar uma simulação com recurso à janela anterior, nesta é possível configurar o **fluxo da água do rio**, o **fator de sedimentação** da água do rio, a **largura** e o **comprimento** do rio, e qual o cenário/**disposição dos sensores** que vai ser simulada.

Assim que a simulação é iniciada, as restantes três janelas aparecem.



Esta janela é a visualização do rio, das **manchas de poluição**, dos **sensores**, e das suas respectivas **cores da coligação a que pertencem** em tempo real.



A janela *Agents energy median*, tal como o título sugere, apresenta um gráfico que é atualizado em tempo real e que representa a **mediana do nível de energia das baterias de todos os sensores** envolvidos na simulação.

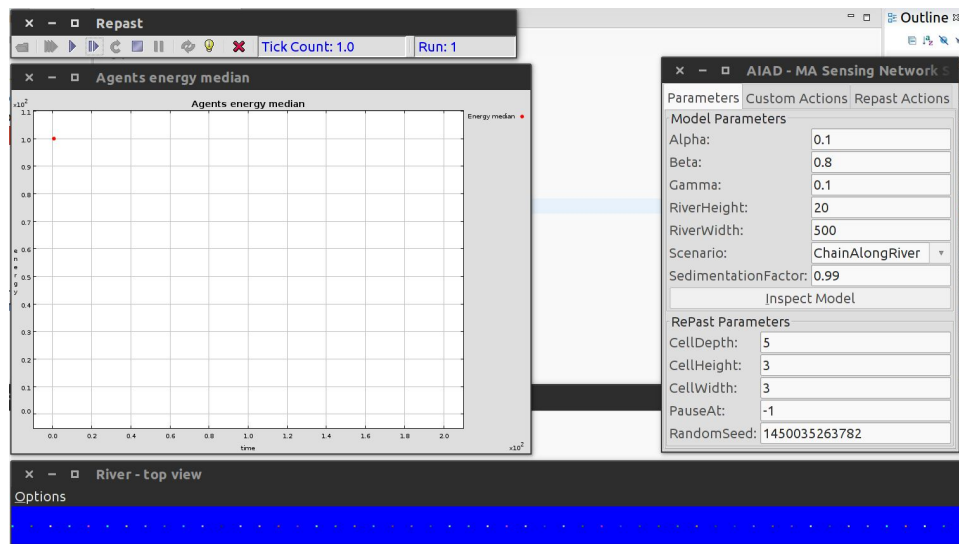
A quinta e última janela é apenas o *output log* da simulação, e por essa razão não é necessário estar aqui exibida.

Experiências

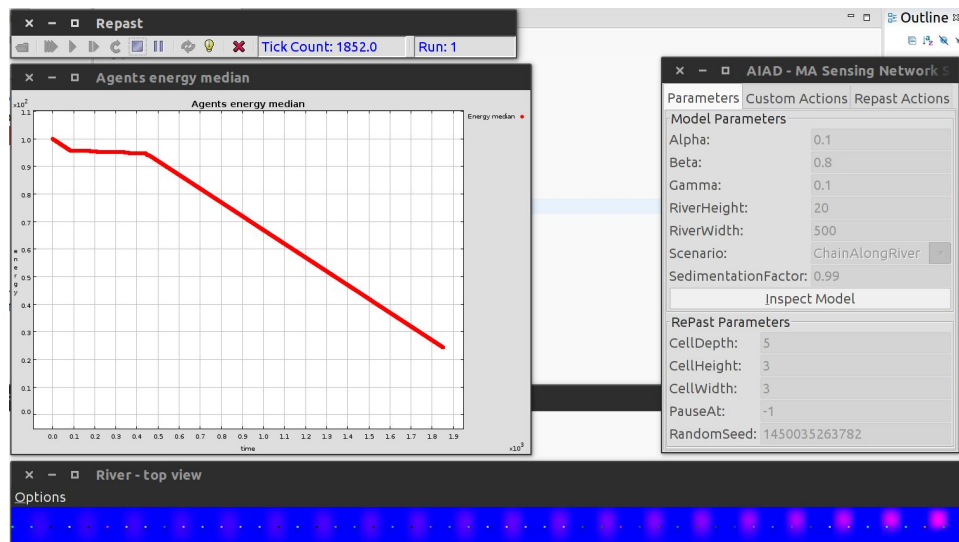
Experiência 1

A primeira experiência que é possível fazer com a aplicação é simular uma disposição em **cadeia linear de sensores** ao longo do rio.

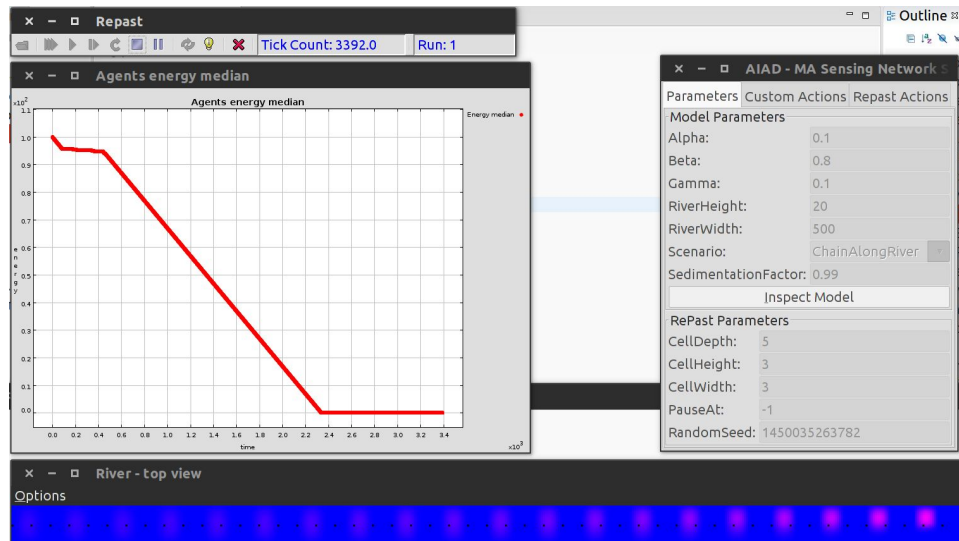
Resultado



Instante inicial da simulação.



Instante intermédio da simulação.

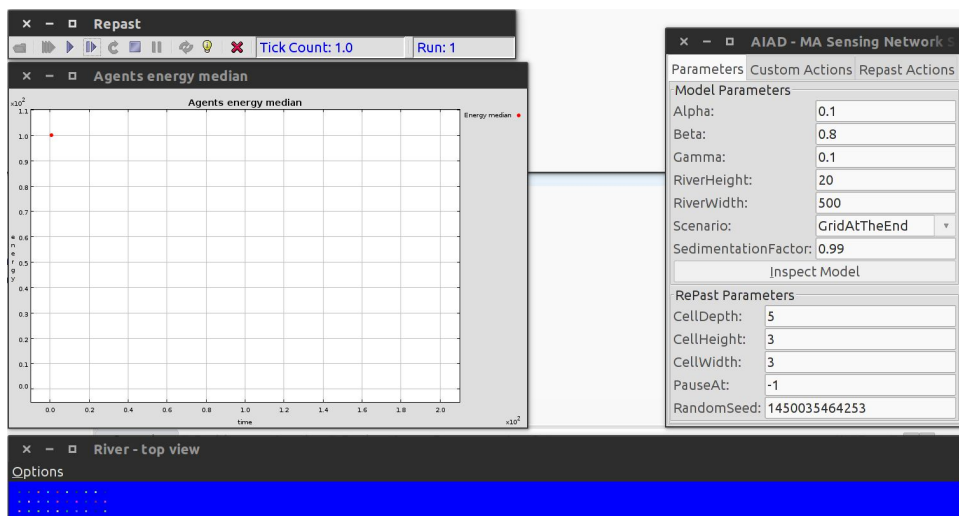


Instante após o final da simulação.

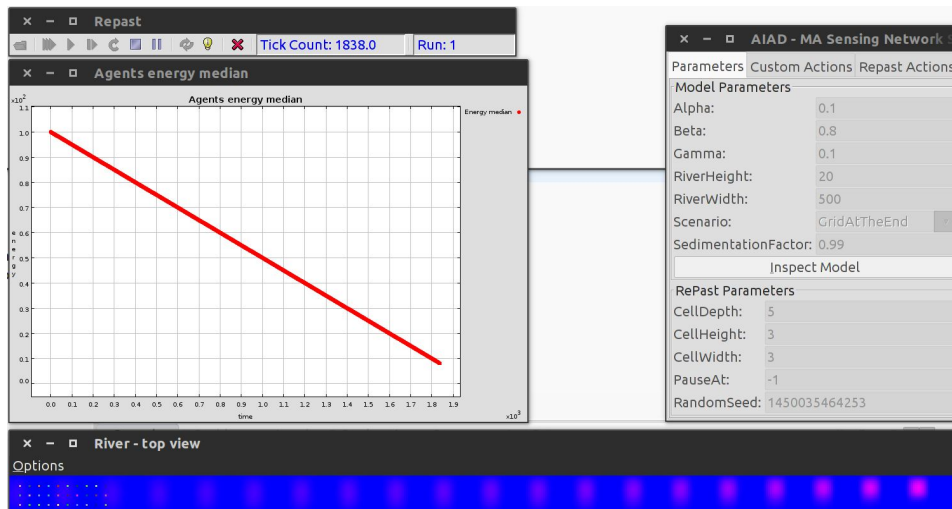
Experiência 2

A segunda experiência que é possível fazer com a aplicação é simular uma disposição de **grelha dos sensores** no final do rio.

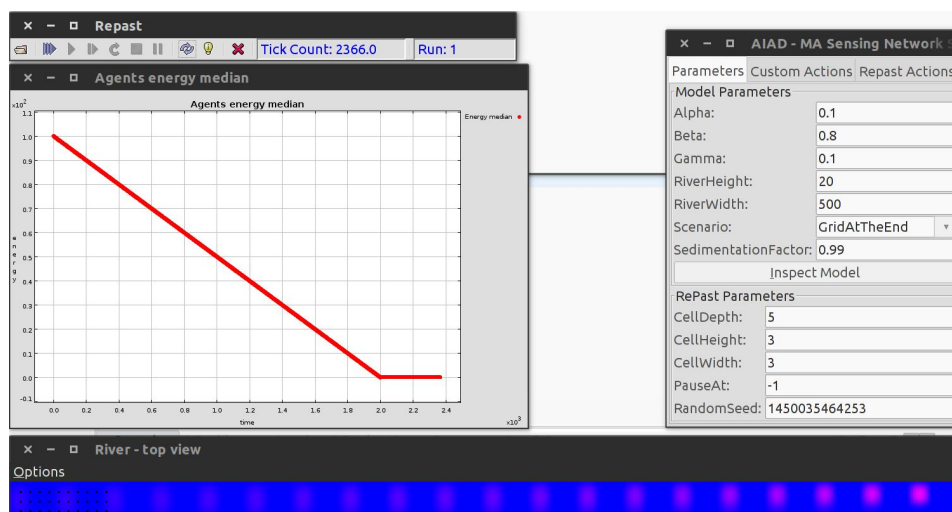
Resultado



Instante inicial da simulação.



Instante intermédio da simulação.

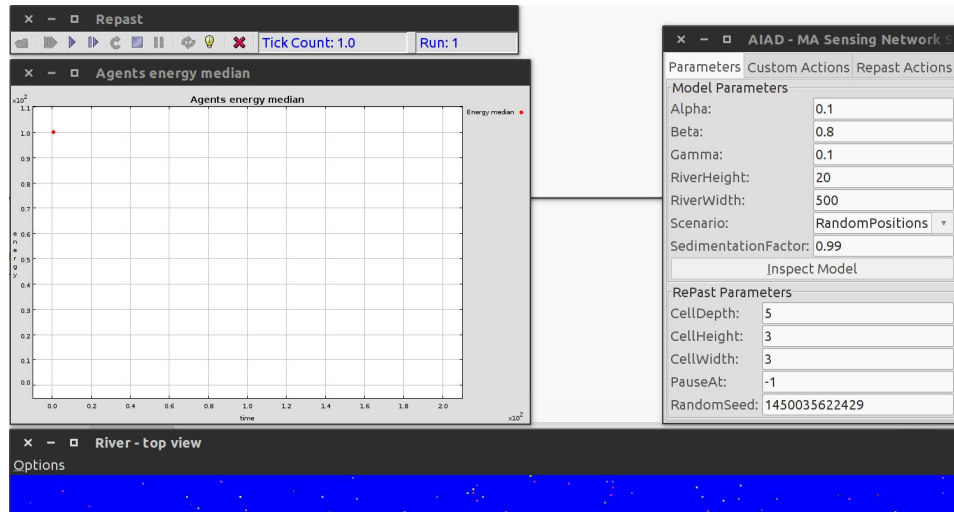


Instante após o final da simulação.

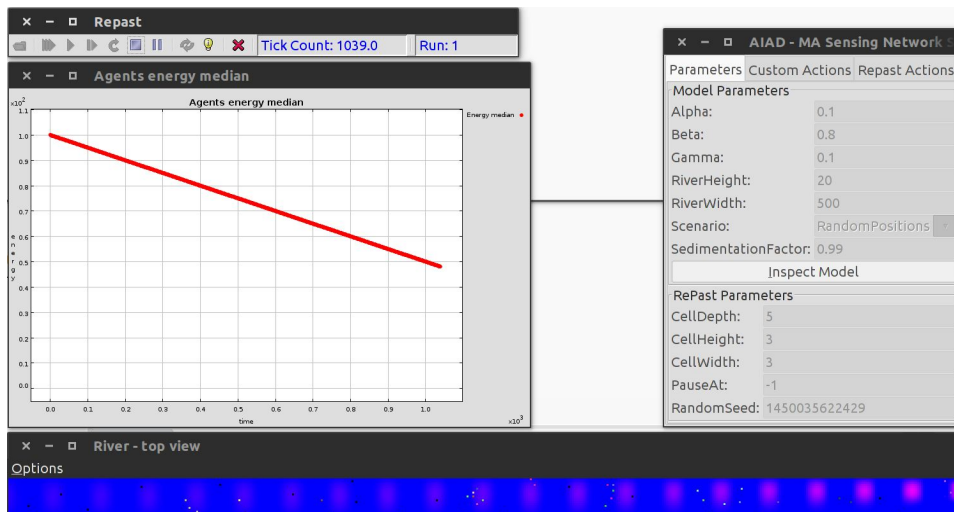
Experiência 3

A terceira experiência que é possível fazer com a aplicação é simular uma **disposição aleatória dos sensores** ao longo do rio.

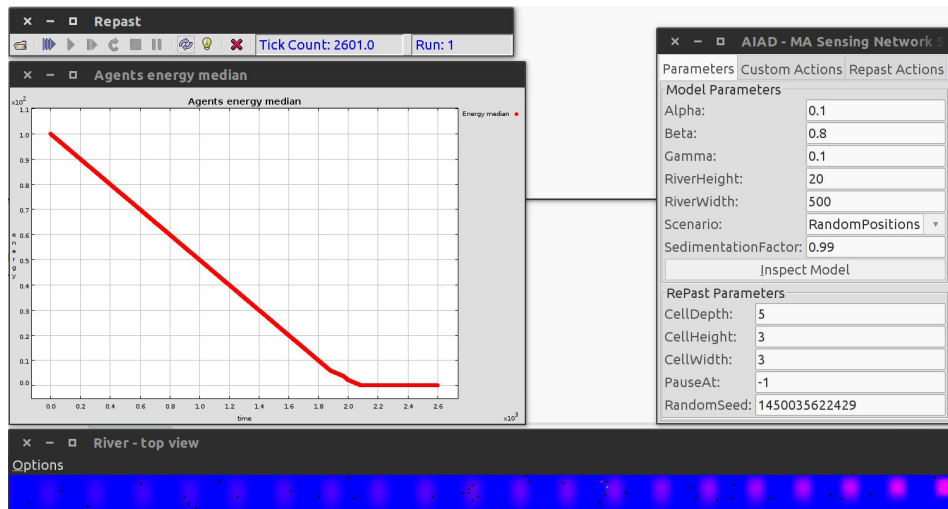
Resultado



Instante inicial da simulação.



Instante intermédio da simulação.



Instante após o final da simulação.

Conclusões

Análise dos resultados das experiências

Depois de correr todas as simulações possíveis, podemos concluir que a disposição que apresenta o **maior tempo de vida global da rede sensorial** é a que usa uma disposição dos sensores em **cadeia linear**. Isso pode ser inferido através da análise do gráfico da mediana dos níveis de energia da bateria de cada sensor da rede.

A **disposição aleatória** é que apresenta o melhor resultado depois da distribuição em cadeia. A **disposição em grelha** no final do rio, não parece trazer qualquer alteração significativa relativamente à poupança de energia.

Desenvolvimento do trabalho

A aplicabilidade de SMA no contexto do caso em estudo é totalmente relevante e pode trazer grandes vantagens ao ser usado em situações concretas da vida real.

O tema é muito interessante e é pertinente para o estudo e implementação dos conceitos lecionados na unidade curricular de AIAD.

Contudo, não podemos deixar de referir e criticar o artigo pelo qual o grupo era suposto basear-se no desenvolvimento do projecto. O artigo carece de muitas explicações detalhadas sobre o protocolo de comunicação entre os agentes, principalmente sobre a forma de avaliação das medidas de *aderência* e *liderança* aquando da formação de uma coligação. Como é que um artigo científico sobre um algoritmo de formação de coligações é capaz de não explicar minuciosamente o processo de formação/negociação de coligações? Toda a mecânica envolvida no algoritmo acenta sobre esse processo.

Por estas e outras razões associadas à qualidade do artigo, o trabalho de desenvolvimento da aplicação por parte do grupo foi extremamente dificultado, e ficou muito aquém das expectativas de cada um dos membros.

Melhoramentos

Devido às lacunas do artigo já mencionadas na secção anterior, há imensos melhoramentos possíveis de serem feitos ao programa desenvolvido pelo grupo.

O grupo fez questão de analisar e tentar melhorar o protocolo proposto no artigo, mas ainda há muito espaço para melhorar este ponto. O protocolo precisa de ser cuidadosamente revisto para garantir que a comunicação assíncrona entre os agentes é correcta, e o menos suscetível a falhas possível.

Outro ponto a melhorar é a representação e visualização das coligações em tempo real. No programa desenvolvido pelo grupo, as coligações podem ser visualizadas através da cor de cada sensor: sensores com cores iguais, fazem parte da mesma coligação. Esta abordagem não permite uma perceção fácil das coligações em tempo real, nem permite saber quem é o líder dessa coligação.

Uma forma possível de melhorar esta falha seria envolver os sensores pertencentes a uma mesma coligação num rectângulo (não preenchido), e atribuir uma cor ao líder distinta da cor dos sensores dependentes do líder dessa coligação.

Para finalizar, um dos melhoramentos a fazer é a implementação do *sink node*. Devido, novamente, à falta de qualidade do artigo, o grupo não teve tempo suficiente para implementar o agente responsável por ser o *sink node*. Este é um sensor que se situa no início do rio, perto da fonte de poluição, e é responsável por registar e guardar as amostragens enviadas pelos líderes.

Este sensor é fundamental porque é com os dados que ele recolhe que é possível analisar a diferença dos valores das amostras que os líderes das coligações lhe fazem chegar, e os valores reais da poluição no rio do modelo de simulação.

Sem este agente implementado, não é possível fazer quaisquer conclusões a respeito da qualidade da informação do algoritmo implementado - apenas é possível tirar conclusões nas poupanças energéticas dos diferentes cenários.

Recursos

Bibliografia

María del Carmen Delgado; Carles Sierra "A Multi-agent Approach to Energy-Aware Wireless Sensor Networks Organization", Agreement Technologies - Second International Conference, AT. Lecture Notes in Computer Science, vol. 8068, Beijing, China, Springer, pp. 32-47, 01/08/2013.

Software

JADE	http://jade.tilab.com/
Repast 3	http://repast.sourceforge.net/repast_3/index.html
SAJaS	https://web.fe.up.pt/~hlc/doku.php?id=sajas
MASSim2Dev	https://web.fe.up.pt/~hlc/doku.php?id=massim2dev
Eclipse	https://eclipse.org/
Ubuntu	http://www.ubuntu.com/

Elementos do grupo

Filipe Ribeiro	10%
Henrique Ferrolho	50%
João Pereira	40%